Setup Kubernetes (K8s) Cluster on AWS

**1. Create Ubuntu EC2 instance - allow 80,443,8080,22**

**2. Update the system using below command**

# apt update -y

**3. install AWSCLI**

# curl https://s3.amazonaws.com/aws-cli/awscli-bundle.zip -o awscli- bundle.zip

# ls -lrt

# apt install unzip python

# unzip awscli-bundle.zip

#./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws

**4. Install kubectl ( Kubectl interact with cluster to manage )**

# curl -LO https://storage.googleapis.com/kubernetes- release/release/$(curl -s https://storage.googleapis.com/kubernetes- release/release/stable.txt)/bin/linux/amd64/kubectl

# chmod +x ./kubectl -761

# sudo mv ./kubectl /usr/local/bin/kubectl

**5. Create an IAM user/role with Route53, EC2, IAM and S3 full access**

Go To iam --> create a Role ----> select EC2 ---> Select EC2 FULL ACCESS , S3 FULL ACCESS , IAM FULL ACCESS , Route53 FULL ACCESS

**6. Attach IAM role to ubuntu server**

**7. Execute aws configure and update the Region**

# aws configure

**8. Install kops on ubuntu instance:**

# curl -LO https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag\_name | cut -d '"' -f 4)/kops-linux-amd64

# chmod +x kops-linux-amd64

# sudo mv kops-linux-amd64 /usr/local/bin/kops

**9. Create a Route53 private hosted zone**

# create an S3 bucket

# aws s3 mb s3://dev.srttraining.online

**10.Expose environment variable:**

# export KOPS\_STATE\_STORE=s3://dev.srttraining.online

**11.Create sshkeys before creating cluster**

# ssh-keygen

**12.Create kubernetes cluster definitions on S3 bucket**

# kops create cluster --cloud=aws --zones=ap-southeast-1b -- name=dev.srttraining.online --dns-zone=srttraining.online --dns private

**13.Create kubernetes cluser**

# kops update cluster dev.srttraining.online --yes

**14.Add the api.internal IP to vi /etc/hosts**

# vi /etc/hosts

**15.Validate your cluster**

# kops validate cluster

**16.To list nodes**

# kubectl get nodes

**17.Deploying Nginx container on Kubernetes**

# Deploying Nginx Container

kubectl run sample-nginx --image=nginx --replicas=2 --port=80

kubectl get pods

kubectl get deployments

Expose the deployment as service. This will create an ELB in front of those 2 containers and allow us to publicly access them:

kubectl expose deployment sample-nginx --port=80 --type=LoadBalancer

kubectl get services -o wide

To delete cluster

kops delete cluster dev.srttraining.online --yes